

# An efficient optimization method to obtain the set of most promising minima in multimodal problems

Álvaro Noriega<sup>1,a</sup>, Ricardo Vijande<sup>1</sup>, Jose Luis Cortizo<sup>1</sup>, Eduardo Rodríguez<sup>1</sup>, Jose Manuel Sierra<sup>1</sup>

<sup>1</sup> Department of Mechanical Engineering, University of Oviedo. Campus de Viesques, 33204 Gijón, Asturias, Spain

Received 13 April 2009, Accepted 27 September 2009

**Abstract** – This paper propounds a new evolution strategy, the Discrete Directions Mutation Evolution Strategy (DDM-ES), with the aim of obtaining the set of most promising minima in multimodal functions and making this process as efficient as possible. First, DDM-ES is compared with a Genetic Algorithm (GA) on two scaleable test functions with 5, 10, 15 and 20 dimensions, showing better behaviour than GA when the objective function is unimodal but not being as global as the GA in highly multimodal ones. Later, the multimodal search nature of DDM-ES is shown applying this ES on two functions with multiple minima. Finally, an application of DDM-ES to the problem of the initial position of a mechanism is shown.

**Key words:** Evolution strategy; Multimodal function; Mechanisms, Initial position problem

## 1 Introduction

The unconstrained optimization problem of a single function (called objective function) can be formulated as a minimization problem as follows:

$$\min f(\mathbf{x}) \quad (1)$$

with  $\mathbf{x} \in \mathcal{R}^n$  and  $\mathbf{x} \in \Omega$

$\Omega$  being the search space in a hyperbox shape. In  $\Omega$ , every variable can take any value between  $\mathbf{l}_i$  and  $\mathbf{u}_i$  with  $i=1, \dots, n$  where  $n$  is the number of variables.

In the search space there can be a single minimum (unimodal function) or several minima (multimodal function). In this latter case, there can be one or several global minima and one or several local minima.

There exist several classifications of the optimization methods [5,17]. Chronologically, the first optimization methods looked to obtain a minimum of the objective function in the most efficient possible way. For do so, certain suppositions about the continuity and derivability of the objective function in the neighbourhood of a minimum are made, producing the mathematical programming methods. These methods [5] need an initial point and they usually obtain the closest minimum to that initial point in a, more or less, efficient way. For this reason, these methods have local behaviour.

Later, and in view of the possibility of more than one minimum existing, the necessity of a minimum search with more global nature arises. This, linked with the fact that in many types of problems the conditions of continuity and derivability cited before cannot be assured and that the derivatives are very expensive to calculate, motivates that the stochastic optimization methods were developed.

These methods consider the objective function as a black box and they try to find its global minimum. Some exam-

ples of this type of method are the simulated annealing (SA) [12], the random search (RS) [9,11] and the evolutive algorithms (EA) [3,13] where the genetic algorithms (GA) [7,10] and the evolutions strategies (ES) [2,18] are the best-known. Some of them have global behaviour (RS and GA) and others, instead, have a more local behaviour (SA and ES).

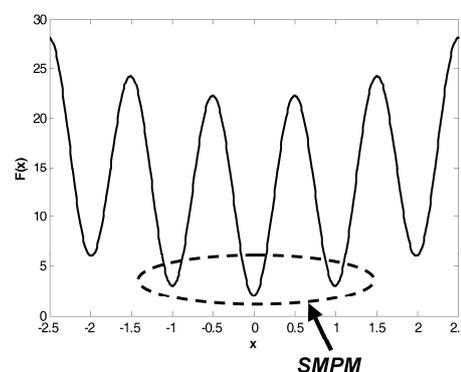


Fig. 1. Example of SMPM

However, many of these methods continue centering the global minimum search, considering it as unique, when, in many problems, the user is interested in knowing if one or several minima of similar values exist. These minima can be called Set of Most Promising Minima (SMPM) and it is composed of all the global minima (in case more than one exists) and the local minima whose values are closer to the global one as well, as is shown in the example of Fig. 1. The size of SMPM depends on the user, who can select the final solution among the solutions contained in this set taking into account additional criteria in a similar way to the Pareto Optimal Front (POF) in multiobjective optimization [1].

The goal of this paper is to propound an optimization method to obtain the SMPM or an approximation of it. This method must:

<sup>a</sup> Corresponding author: [noriegaalvaro@uniovi.es](mailto:noriegaalvaro@uniovi.es)

a) Differentiate between the different minima allowing the knowledge of the user about the topology of the objective function to be increased and offering more possibilities to select the final solution.

b) Make the process as efficient as possible.

The paper is organized as follows: In Section 2, the new optimization method, called Discrete Directions Mutation Evolution Strategy (DDM-ES), is described. In Section 3, the results of two benchmark tests between the DDM-ES and a GA are shown. They show that DDM-ES has the desired features a) and b). In Section 4, a real application with multiple global solutions is shown and DDM-ES is used to solve it in a satisfactory way. In Section 5, a set of conclusions are extracted about the new method proposed.

## 2 Description of the DDM-ES

The Discrete Directions Mutation Evolution Strategy [15,16] is an evolution strategy of type  $(\mu + \lambda) - ES$  which is characterized by its mutation operator.

This mechanism of mutation consists of adding to the vector that represents the parent  $\mathbf{x}_{parent}$ , a vector  $v \cdot \mathbf{d}$  that represents the mutation. The unitary vector  $\mathbf{d}$  represents the direction of mutation in the n-dimensional space and  $v$  is the mutation step:

$$\mathbf{x}_{offspring} = \mathbf{x}_{parent} + v \cdot \mathbf{d} \quad (2)$$

Moreover, the DDM-ES has the following additional specific features.

### 2.1 Specific features

#### Workspace

The population  $\mathbf{P}$  is made by a set of individuals represented by vectors  $\mathbf{x}'$ , which are defined in a workspace where every variable  $x'_i$  belongs to the interval [0,1].

When an individual  $\mathbf{x}'$  must be evaluated, it must be scaled first to the search space where the objective function is defined using the following expression applied to each component:

$$\mathbf{x}_i = \mathbf{l}_i + (\mathbf{u}_i - \mathbf{l}_i) \cdot \mathbf{x}'_i \quad (3)$$

where  $\mathbf{x}$  is the vector that represents the individual  $\mathbf{x}'$  in the search space,  $\mathbf{l}$  is the vector that contains the lower bound of the variables and  $\mathbf{u}$  is the vector with the upper bound of the variables.

The vector  $\mathbf{d}$  which represents the direction of mutation is defined in the workspace.

#### Discrete directions of mutation

This is the most important feature of this method and the one that gives it its name (DDM-ES).

The classical method to generate unitary n-dimensional vectors with uniform distribution in the space consists of randomly generating points inside a hypercube according to a uniform distribution of probability. The vector that joins the centre of the hypercube with the generated point defines the direction of the unitary vector. Only the points generated inside the hypersphere inscribed in the hypercube are valid and, then, every direction has the same probability of

being selected. The drawback is that the hypercube's volume (and, therefore, the probability of generating individuals in its volume) is concentrated in the vertexes when the number of dimensions increases as can be seen in Table 1 where the quotient between the volumes of the hypercube and the hypersphere inscribed on it is shown when the number of dimensions increases. Thus, the probability of generating individuals in the hypersphere tends to 0 when the number of dimensions increases. Although there exist some methods to generate random samples in and on a hypersphere [8,14], there is not an efficient method to generate n-dimensional unitary vectors with randomly and uniformly distributed directions.

Table 1. Variation of volume quotient

Number of dimensions	$V_{hypersphere} / V_{hypercube}$
1	1
2	0,7854
3	0,5236
4	0,3084
5	0,1645
6	0,0807

For this reason, it is proposed to discretize the infinite possible directions to a finite set that allows easy and efficient generation and that, furthermore, these directions are distributed in quite a homogeneous way over the hypersphere's volume.

To carry out this discretization the following method is proposed. First of all, a unit hypercube is generated with the individual to mutate situated in its centre. Secondly, a face of the hypercube is randomly selected and on this face is generated a grid of equally-spaced nodes. Then, one node of this grid is randomly selected. Finally, the vector that joins the hypercube's centre with the selected point indicates the mutation direction. A graphic representation of this method can be seen in Fig. 2.

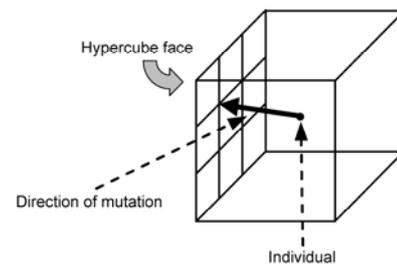


Fig. 2. Discrete direction generation

The number of nodes in the grid generated in the hypercube's face is controlled by means of a parameter of the method called  $n_{dv}$  which allows the discretization (number of possible directions for mutation) to be controlled. The bigger this parameter is, the smaller the homogeneity of the distribution of the directions is.

#### Mutation step

As is indicated at the beginning of this section, the mutation step  $\nu$  is unique and it is randomly generated according to a normal distribution with mean 0 and standard deviation  $\sigma$  (called mutation strength) for every offspring. Both of them,  $\nu$  and  $\sigma$ , are always referred to the workspace.

#### Dynamic control of mutation strength

The mutation strength decreases with the generations according to an expression fixed by the user of the algorithm. This idea is similar to the one used in simulated annealing to modify the temperature. The expression selected for the DDM-ES is the following:

$$\sigma_j = \left(1 - \frac{R}{100}\right) \cdot \sigma_{j-1} \quad (4)$$

where  $j$  is the number of the current generation and  $R$  is the percentage decrease of the mutation strength from a generation  $\sigma_j$  with regard to the previous generation  $\sigma_{j-1}$ . This expression allows the automatic advancing of several generations or advancing from one generation to the next one manually.

#### Valid and non valid individuals

Inside a population of offsprings generated by mutation from their parents, the ones that belong to the workspace are called valid individuals, and the ones that are outside the workspace, non-valid individuals. The appearance of non-valid individuals is possible due to the fact that the mutation step  $\nu$  is generated according to a normal distribution which is defined between  $-\infty$  and  $+\infty$ , and as a result, the offspring can be outside the workspace.

#### Independent individuals

There also exists another type of individual, called independent individuals, which are generated in the workspace according to a uniform distribution of probability.

The independent individuals have a mission to provide global convergence to the DDM-ES since they make a pure random search. Their effect is similar to the mutation in a GA.

In DDM-ES, the population of offsprings (individuals generated by mutation and independent individuals) has a constant size  $\mu$ , but the amount of individuals of every type is variable depending on the search progress. The number of independent individuals of every generation increases (parameter  $\Delta_{ii}$ ) if the search is stagnated and is reduced to a minimum (parameter  $n_{ii}$ ) when the search obtains an improvement with regard to the previous generation.

The initial population is generated with  $\mu$  independent individuals.

## 2.2 Pseudocode

The detailed pseudocode of the DDM-ES is shown below:

**Input:**  $\mu$  (Population size)  
 $g$  (Number of generations)

$\sigma_1$  (Initial mutation strength)  
 $R$  (% decrease of  $\sigma$  in every generation)  
 $n_{ii}$  (Initial independent individuals)  
 $\Delta_{ii}$  (Increase of independent individuals)  
 $n_{dv}$  (Divisions per variable)  
 $\mathbf{l}, \mathbf{u}$  (Vectors with lower and upper bounds)

**Output:**  $m$  (Optimal individual)  
 $fm$  (Optimal individual evaluation)

**Step 1:** Initialization: Set the number of variables  $n$ . Generate initial population  $\mathbf{P}_0$ . Initialize the number of independent individuals  $n_{indep} = n_{ii}$ .

**Step 2:** Evaluation and saving: Evaluate  $\mathbf{P}_0$  to obtain  $\mathbf{F}_0 = f(\mathbf{P}_0)$ . Initialize  $m$  and  $fm$  with the best individual found in  $\mathbf{F}_0$ .

**Step 3:** Main loop: For  $j = 1$  to  $j = g$

a) Calculate  $\sigma_j$

b) Generate offspring population  $\mathbf{P}_{offspring}$

Repeat  $\mu - n_{indep}$  times:

Random selection of a parent  
 Random generation of a mutation direction  
 Generate an offspring by mutation of the parent  
 If offspring is non valid:

Replace offspring with an independent individual

c) Evaluate  $\mathbf{P}_{offspring}$  to obtain  $\mathbf{F}_{offspring}$

d) Sorting  $\{\mathbf{F}_{j-1}, \mathbf{F}_{offspring}\}$  and  $\{\mathbf{P}_{j-1}, \mathbf{P}_{offspring}\}$

e) Selection of the  $\mu - n_{indep}$  best offsprings of reordered  $\{\mathbf{P}_{j-1}, \mathbf{P}_{offspring}\}$  to obtain  $\mathbf{P}_{best}$  and  $\mathbf{F}_{best}$

g) Generate and evaluate indep. individuals to obtain  $\mathbf{P}_{indep}$  and  $\mathbf{F}_{indep}$

h) Build the new population:  $\mathbf{P}_j = \{\mathbf{P}_{best}, \mathbf{P}_{indep}\}$  and  $\mathbf{F}_j = \{\mathbf{F}_{best}, \mathbf{F}_{indep}\}$

i) Obtain the best individual  $m_{new}$  and its evaluation

$fm_{new}$

j) Update  $n_{indep}$

if  $fm = fm_{new}$  and  $n_{indep} < \mu - 1$

$n_{indep} = n_{indep} + \Delta_{ii}$

if  $\mu - 1 < n_{indep}$

$n_{indep} = \mu - 1$

else if  $fm_{new} < fm$

$n_{indep} = n_{ii}$

## 2.3 Parameter setting

The population size  $\mu$  is chosen taking into account the possible multimodality of the objective function. The bigger the number of minima expected in the objective function is, the bigger the population size must be. It is also possible to relate the population size to the number of dimensions of the search space  $n$ . Based on the test made by the authors, the empirical expression  $\mu = 50 \cdot n$  is proposed.

The number of generations  $g$  and the value of the percentage decrease  $R$  between generations affect to the variation of the mutation strength  $\sigma$ . Based on several tests made, the recommended values are  $R = 5$  and  $g = 50$  which make the DDM-ES converge to the minima very quickly unless the function is very complex.

For the mutation strength of the first generation  $\sigma_1$ , the following expression is proposed:

$$\sigma_1 = \frac{\sqrt{n}}{2} \cdot \left( \frac{1}{\mu} \right)^{\frac{1}{n}} \quad (5)$$

This expression arises from considering the hypervolume of the workspace (whose value is 1) equitably shared among the  $\mu$  individuals of the population and supposing that the hypervolume corresponding to each individual adopts the shape of a hypercube. Then, the initial mutation strength is the distance between the centre of the hypercube and one vertex of it. A graphical example of this idea can be seen in Fig. 3.

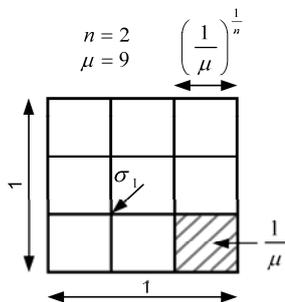


Fig. 3 Graphical example of the meaning of  $\sigma_1$

The number of initial independent individuals  $n_{ii}$  must be set as a small value and thus not decrease the exploitation ability of the algorithm. A recommended value is 0. The increase in the number of independent individuals when the evolution is stagnated  $\Delta_{ii}$  will be a small value compared with  $\mu$ , for instance, 5%. When you do not want to hinder the convergence to the minima found, the value of  $\Delta_{ii}$  is set to 0.

The number of divisions of the grid per variable  $n_{dv}$  used in the test made is 3 and no obvious improvement is observed when this value increases.

### 3 Benchmarking

In this section, the outcomes of two benchmark tests between the DDM-ES and a GA are shown. The platform used was Matlab® where the code of DDM-ES is programmed.

Moreover, the Genetic Algorithm and Direct Search Toolbox that implements a highly adjustable GA was used.

#### 3.1 Test 1

In this test, it is intended to compare the performance, as optimization algorithms, of the DDM-ES and the GA when they have the same computational cost.

##### *Objective functions and conditions*

For this test, two standard and scaleable test functions are selected. Their features are shown in Table 4.

The main difference between both functions is the total number of minima. While the hyper-sphere function is unimodal, the Rastrigin's function is highly multimodal and its number of minima increases with the number of variable ( $11^n$  minima), increasing its complexity a great deal.

5, 10, 15 and 20 variables are used with every test function. The population size and the number of generations are the same with every combination of test function and number of variables. The test will be repeated 30 times. In each repetition, the initial population is randomly generated outside the algorithm and it will be common to both of them.

Both algorithms evolve from the same initial population for the same number of generations and the mean evolution of the minima found in every generation is extracted. This evolution will be shown by means of a semilogarithmic graph to compare the accuracy and the value of the minima reached with each algorithm. As the two test functions selected have a global minimum of value 0, the value of this function coincides with the absolute error and it can also be used to verify if an algorithm converges to the global minimum or to a local minimum.

The runtime is also extracted with every test function with both algorithms and in every repetition to calculate the mean of the runtime for the 30 repetitions.

The number of generations  $g$  for both algorithms will be 100 and the population size  $\mu$  and the initial mutation strength  $\sigma_1$  will be the ones indicated in 2.3.

The specific parameters used with the DDM-ES are shown in Table 2 and the specific parameters for the GA are shown in Table 3.

Table 2. Parameters for DDM-ES

Parameter	Value
$\mu$	$50 \cdot n$
$g$	100
$\sigma_1$	(see Section 2.3)
$R$	5
$n_{ii}$	0
$\Delta_{ii}$	$n$
$n_{dv}$	3

The ga\_out function allows the best value obtained on each generation to be externally saved. The computer used for this test has a processor Intel® Pentium® IV, 3.0 GHz and 1 GB of RAM.

Outcomes and discussion

The outcomes of the benchmarking on the function hyper-sphere are shown in Fig. 4. On it, it can be seen that the DDM-ES reaches the minimum with higher precision than the GA in all cases. Observing the evolution of accuracy with the generations, the DDM-ES improves the accuracy

according to a law that can be set to a logarithmic equation while the GA obtains a faster initial improvement of the accuracy. However, later, this improvement slows down and the DDM-ES always finishes by overcoming the GA in accuracy.

Table 4. Test functions

Name	Expression	Global minimum		Features
		Location	Value	
Hyper-sphere	$f(\mathbf{x}) = \sum_{i=1}^n \mathbf{x}_i^2$ $-5.12 \leq \mathbf{x}_i \leq 5.12$	$\mathbf{x}_i = 0$	0	Unimodal
Rastrigin	$f(\mathbf{x}) = 10 \cdot n + \sum_{i=1}^n (\mathbf{x}_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot \mathbf{x}_i))$ $-5.12 \leq \mathbf{x}_i \leq 5.12$	$\mathbf{x}_i = 0$	0	Multimodal
MGM	$f(\mathbf{x}) = \sum_{i=1}^n (\mathbf{x}_i^2 - 2)^2$ $-2.5 \leq \mathbf{x}_i \leq 2.5$	$\mathbf{x}_i = \pm 2$	0	Multimodal

Table 3. Parameters for GA

Population		Reproduction	
Population type	Double	Elite count	2
Population size	$50 \cdot n$	Crossover fraction	0,8
Initial population	-	<b>Migration:</b>	None
Bounds	-	<b>Algorithm settings:</b>	By default
Fitness Scaling		Hybrid function: None	
Scaling function	Proportional	Stopping criteria	
Selection		Generations	100
Selection function	Roulette	Time limit	Inf
Mutation		Fitness limit	-Inf
Mutation function	Gaussian	Stall generations	100
Scale	1	Stall time limit	3600
Shrink	1	Function tolerance	1e-6
Crossover		Output function: ga_out	
Crossover function	Single point	Vectorized: Off	

The outcomes of the benchmarking on the Rastrigin’s function are shown in Fig. 5. In the case of 5 variables, the GA is able to find the global minimum while the DDM-ES obtains a nearby local minimum. In the other cases, both algorithms obtain a local minimum, with the minimum obtained by the GA always being better.

In Table 5 the mean runtimes are shown and you can notice that the DDM-ES is faster than the GA in all cases. As the number of function evaluations is the same for both algorithms, you can conclude that the DDM-ES has an internal computational cost smaller than the GA. The improvement in runtime of the DDM-ES over the GA increases when the number of variables increases and it approximately varies between 25% with 5 variables and 57% with 20 variables.

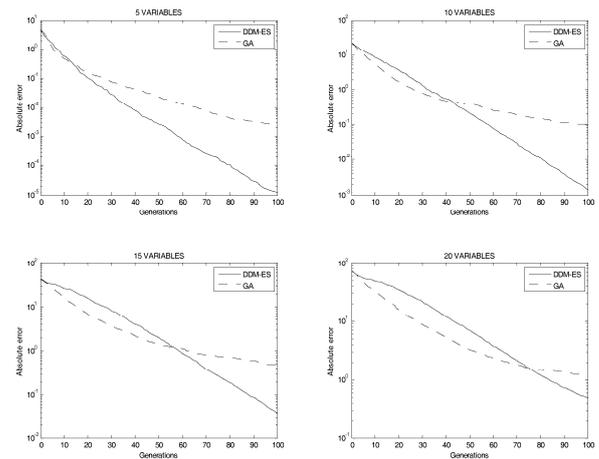


Fig. 4 Outcomes of Hyper-sphere function

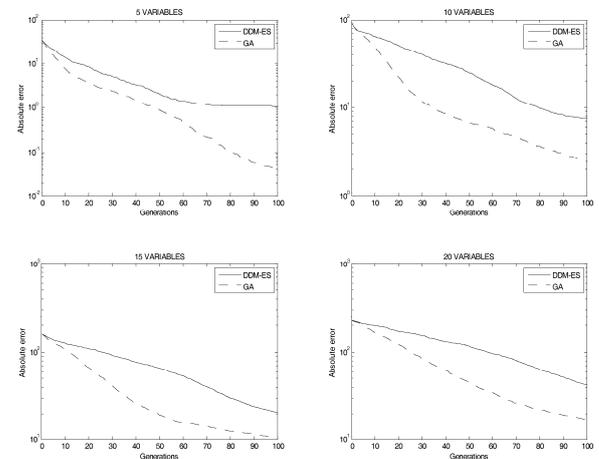


Fig. 5 Outcomes of Rastrigin’s function

Table 5. Mean runtime (s)

Function	Algorithm	Number of variables			
		5	10	15	20
Hyper-sphere	DDM-ES	3.4895	8.5762	16.6997	26.6708
	GA	4.6739	13.4234	31.6905	61.9313
Rastrigin	DDM-ES	3.5729	8.6705	16.5921	26.9703
	GA	5.0889	15.8101	32.1995	54.7577

### 3.2 Test 2

In this test, it is intended to compare the computational cost needed to obtain the SMPM with enough definition and accuracy.

#### *Objective function and conditions*

In this case, two test functions have been selected: the Rastrigin's function with 2 variables and the function MGM (Multiple Global Minima) with four variables. This last function (proposed by the authors of this paper) is scaleable, multimodal and it has the special feature that its  $2^n$  minima are all global. Its features are shown in Table 4.

One run will be made with every optimization method (DDM-ES and GA) from the same initial population of size 200 and evolving for 50 generations. The other parameters used will be the ones indicated in Test 1 except for the parameter  $\Delta_{ii}$  which is set to 0 so as not to hinder the convergence to the minima already found and to improve the definition which you can observe the graphical representation of this process.

To compare both algorithms, the populations of some generations will be shown. In the case of Rastrigin's function, the individuals will be directly shown in the search space. In the case of function MGM, the individuals will be shown in the workspace using a representation by means of parallel coordinates similar to the one used in the decision space of multiobjective optimization [4].

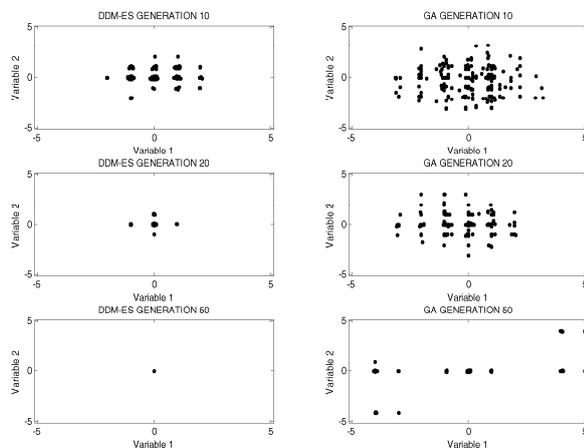


Fig. 6 Populations with Rastrigin's function with 2 variables

#### *Outcomes and discussion*

The outcomes of the Rastrigin's function are shown in Fig. 6. In the three generations shown, the population of the DDM-ES is more crowded than the population of the GA, defining with more clarity the location of the minima. In the case of the DDM-ES, it is noticed that the population is concentrated on the global minimum and on the minima located in its neighbourhood, that is to say, the SMPM. The set of minima that makes up the SMPM in every generation decreases monotonously with the generations since the global minimum is alone. In the case of the GA, there exists a decrease in the number of minima defined by the population but they do not concentrate around the global minimum.

Moreover, it is noticed that the DDM-ES converges to the minima in a more efficient way than the GA. For instance, the DDM-ES in generation 20 shows a set of 5 minima with a high degree of definition while the GA in the generation 50 still has 12 minima with a similar definition but with more spreading.

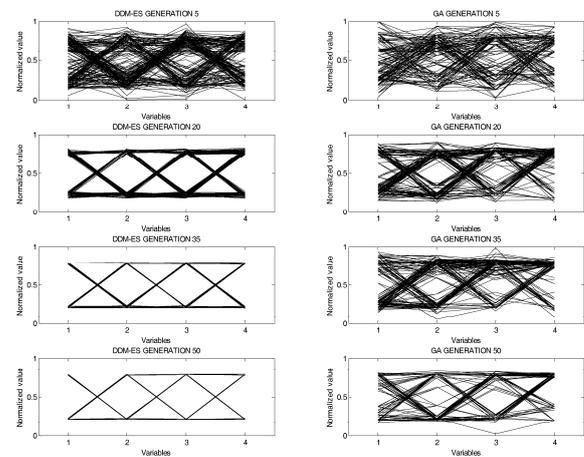


Fig. 7 Populations with function MGM with 4 variables

The outcomes of the test on the function MGM are shown in Fig. 7. In this figure, you can see that the DDM-ES converges to the 16 global minima (depending on the generation) with more definition and efficiency since in the generation 20, the DDM-ES has clearly obtained the location of all the minima of the function while the GA does not obtain the same definition of the minima even in the generation 50 with a computational cost 2.5 times greater. Furthermore, with the passing of generations, the DDM-ES refines the minima found since the strip around every minimum become tighter and it means that the individuals are more grouped around the minimum and, therefore, they represent more precise solutions. This refining process is slower with the GA.

Looking at these outcomes, it is possible to state that the DDM-ES makes a more efficient use of the computational effort made than the GA, supplying a more refined SMPM as well.

## 4 Application to the problem of initial position of a mechanism

### 4.1 Problem description

The problem is about determining all the initial positions that the mechanism of six bars of Watt (shown in Fig. 8)

can have when the value of its degree of freedom  $\alpha$  is set. Every possible initial position has caused a different kinematic simulation of the mechanism.

The lengths of the bars are known and the locations of the joint with the frame too. All these parameters are shown in Table 6.

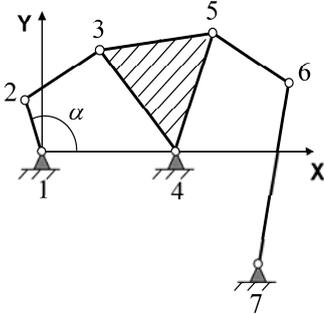


Fig. 8 Mechanism of six bars of Watt

Table 6. Dimensional parameters

$x_1 = 0$	$L_{12} = 2$
$y_1 = 0$	$L_{23} = 5$
$x_4 = 5$	$L_{34} = 5$
$y_4 = 0$	$L_{35} = 5$
$x_7 = 9$	$L_{45} = 6$
$y_7 = -5$	$L_{56} = 4$
$\alpha = 120^\circ$	$L_{67} = 8$

This mechanism can be modeled by means of natural coordinates [6] generating the following constraint equations:

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 - L_{12}^2 = 0 \quad (6)$$

$$(x_2 - x_3)^2 + (y_2 - y_3)^2 - L_{23}^2 = 0 \quad (7)$$

$$(x_3 - x_4)^2 + (y_3 - y_4)^2 - L_{34}^2 = 0 \quad (8)$$

$$(x_3 - x_5)^2 + (y_3 - y_5)^2 - L_{35}^2 = 0 \quad (9)$$

$$(x_4 - x_5)^2 + (y_4 - y_5)^2 - L_{45}^2 = 0 \quad (10)$$

$$(x_5 - x_6)^2 + (y_5 - y_6)^2 - L_{56}^2 = 0 \quad (11)$$

$$(x_6 - x_7)^2 + (y_6 - y_7)^2 - L_{67}^2 = 0 \quad (12)$$

$$(x_2 - x_1) - L_{12} \cdot \cos \alpha = 0 \quad (13)$$

$$(y_2 - y_1) - L_{12} \cdot \sin \alpha = 0 \quad (14)$$

The 9 constraint equations can be represented in a more compact way as:

$$\Phi(\mathbf{q}) = \mathbf{0} \quad (15)$$

$\Phi(\mathbf{q})$  being a column vector of 9 components where the left sides of the constraint equations are stored and  $\mathbf{q}$  is the

vector where the cartesian coordinates of the mobile nodes of the mechanism are stored:

$$\mathbf{q} = [x_2 \ y_2 \ x_3 \ y_3 \ x_5 \ y_5 \ x_6 \ y_6] \quad (16)$$

The non-linear system of equations shown can usually be solved using Newton's method, which needs an initial approximation and it obtains one solution.

However, this system of equations can have more than one solution. To determine all the existing solutions, the problem of finding the roots can be converted in an error minimization problem where the error measures the simultaneous fulfilment of the constraint equations.

The error made by a vector of coordinates  $\mathbf{q}$  is defined as:

$$error(\mathbf{q}) = \sum_{i=1}^9 \Phi_i^2(\mathbf{q}) \quad (17)$$

Those solutions  $\mathbf{q}$  that make the error 0 will be the solutions of the non-linear system of equations posed.

## 4.2 Parameters used with DDM-ES

The lower and upper bounds of the variables are defined, depending on the type of coordinate, in the following way:

$$\begin{aligned} x &\in [-5, 15] \\ y &\in [-10, 10] \end{aligned} \quad (18)$$

The parameters used with the DDM-ES are shown in Table 7.

Table 7. Parameters of DDM-ES

Parameter	Value
$\mu$	1000
$g$	50
$\sigma_1$	0.5964
$R$	5
$n_{ii}$	0
$\Delta_{ii}$	0
$n_{dv}$	3

## 4.3 Filtering and refinement of solutions

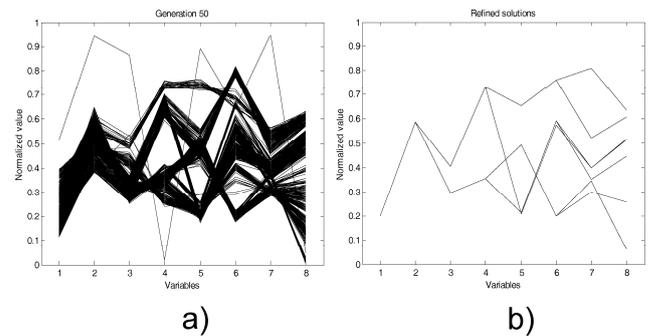


Fig. 9 Population in generation 50 and filtered and refined solutions

After running the DDM-ES with the parameters indicated before, the final population shown in Fig. 9.a is obtained.

The individuals of this population must be filtered to eliminate redundant individuals. During the filtering process, the individuals closer to one, which acts as a reference, and which have greater error than the reference individual are eliminated.

The individuals which have passed the filter are used as initial approximations for a deterministic optimization method of second order. The method used in this application is a quasi-Newton method that uses the BFGS technique to update the Hessian matrix [5]. The individuals of the population that have passed the processes of filtering and refining can be seen in Fig. 9.b.

The refining process of the filtered individuals has a very low cost because the initial approximations are close to the final solution (they are the solutions obtained by the DDM-ES) and because this method is very efficient.

#### 4.4 Discussion

The filtered and refined population offers, with high precision, 7 of the 8 possible solutions for the problem of initial solution of the mechanism. As an example, two of these solutions, which correspond to two possible configurations of the mechanism, which furthermore, are far away from each other, can be seen in Fig. 10.

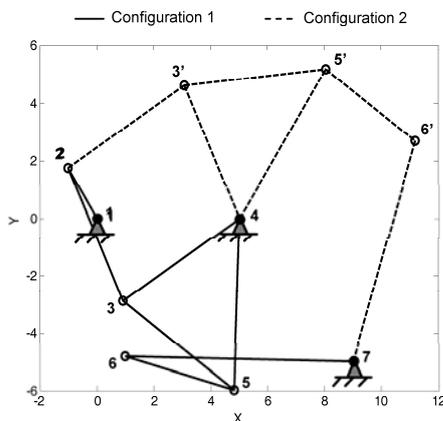


Fig. 10 Two possible configurations for the mechanism

#### 5 Conclusions

The evolution strategy called DDM-ES and proposed in this paper constitutes an optimization method which shows greater precision than a GA with the same computational cost but it does not show a behaviour as global as the GA on highly multimodal functions.

However, the DDM-ES allows an approximation of the set of most promising minima (SMPM) to be obtained. This set monotonously reduces its size, eliminating the less promising minima during the successive generations because its visual monitoring by the user is very intuitive. Moreover, these minima are defined with greater precision than with the GA. Finally, the obtaining of SMPM is more efficient than with the GA.

This feature contributes to obtaining a better knowledge by the user about the interesting part of the objective func-

tion's topology when it has several minima, allowing the user to make his choice among several optima according to criteria not specified in the problem definition as sensitivity of the minima or fulfilment of constraints not specified before.

The DDM-ES has been applied to the problem of obtaining the initial position of a mechanism, which can be solved before making its kinematic or dynamic simulation. This problem is multimodal and it has several global minima.

For the practical solving of this problem, the DDM-ES has been modified by adding a stage of filtering to eliminate the redundant individuals of the population and later a stage of refining the filtered individuals. In this case, the DDM-ES allows almost all the solutions of the problem to be obtained with high precision.

#### References

1. J. Andersson, *A survey of multiobjective optimization in engineering design*, Technical Report LiTH-IKP-R-1097, Sweden (2000)
2. T. Bäck, F. Hoffmeister, H.P. Schwefel, A Survey of Evolution Strategies, In *Proceedings of the Fourth International Conference on Genetic Algorithms* (Los Altos, California, USA, 1991)
3. T. Bäck, *Evolutionary algorithms in theory and practice*, (Oxford University Press, 1996)
4. C.M. Fonseca, *Multiobjective Genetic Algorithms with Applications to Control Engineering Problems*, PhD Thesis, University of Sheffield (1995)
5. R.L. Fox, *Optimization Methods for Engineering Design*, (Addison-Wesley, Reading, Massachusetts, 1971)
6. J. García de Jalón, E. Bayo, *Kinematic and dynamic simulation of multibody systems. The real-time challenge*, (Springer-Verlag, New York, 1994)
7. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, (Addison-Wesley, Reading, Massachusetts, 1989)
8. G. Guralnik, C. Zemach, T. Warnock, *An algorithm for uniform random sampling of points in and on a hypersphere*, Inf. Process. Lett. 21(1), 17-21 (1985)
9. C. Hamzaçebi, F. Kutay, *A heuristic approach for finding the global minimum: Adaptive random search technique*, Appl. Math. Comput. 173(2), 1323-1333 (2006)
10. J.H. Holland, *Adaptation in Natural and Artificial Systems*, (MIT Press, Cambridge, Massachusetts, 1975)
11. J. Jezowski, R. Bochenek, G. Ziomek, *Random search optimization approach for highly multi-modal nonlinear problems*, Adv. Eng. Softw. 36, 504-517 (2005)
12. S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, *Optimization by Simulated Annealing*, Science 220, 671-680 (1983)
13. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, (Springer-Verlag, 1992)
14. J. Nie, B.R. Ellingwood, *Directional methods for structural reliability analysis*, Struct. Saf. 22, 233-249 (2000)
15. A. Noriega, E. Rodríguez, J.L. Cortizo, R. Vijande, J.M. Sierra, *A new evolution strategy for the unconstrained optimization problem*. In *Proceedings of Second International Conference on Multidisciplinary Design Optimization and Applications*, (Gijón, Spain, 2008)
16. A. Noriega, *Síntesis dimensional óptima de mecanismos mediante estrategias evolutivas*, PhD. Thesis, University

of Oviedo (2008)

17. S.S. Rao, *Engineering Optimization, Theory and Practice*, (John Wiley & Sons, New York, 1996)
18. H.P. Schwefel, *Evolution and Optimum Seeking*, (John Wiley & Sons, New York, 1995)